COMS 363 Section C Assignment II

Fall 2018

Date assigned: 9/6/2018
Due date: 9/17/2018 by 11:50pm
Percentage in your final grade: 5%
Maximum score for the assignment: 100 points

Objectives

1. Practice mapping entities, relationships, and constraints from an ER diagram to tables (relations) in the relational data model.
2. Practice Data Manipulation Language (DML) and Data Definition Language (DDL) of Structure Query Language (SQL) in MySQL.

Instructions:

The assignment is individual work, not a collaborative work. Compress all the answers into one zip file. Name the file <netid>HW2.zip where <netid> is replaced by your netid.

Questions

1. (20 points) Given the ER diagram in Figure 1 below, map the entity sets, relationship sets, and the constraints into relations and integrity constraints. Your mapping should minimize the number of relations created while enforcing all the constraints in the ER diagram.

   For each relation, specify the relation name, the attribute name and type, the primary key attribute, any unique attribute, and any foreign key to which referenced relation. See the example given as in the class activity on September 4, 2018.

2. (35 points) Provide SQL DDL statements to create a relational database with all the relations and integrity constraints in your answer for Q1. For any referential integrity constraints, use ON DELETE CASCADE ON UPDATE CASCADE as the option. Put all these SQL statements in <yournetid>HW2_Q2.sql where <yournetid> is replaced with your actual netid, for instance, tavanapoHW2_Q2.sql.

   - Make sure that your DDL statements produce correct results in MySQL.
   - Use your netid as your database name.
   - Make sure your DDL drops the database if it exists before creating the new database.
   - Make sure to enforce all the constraints using the schema design and only when the schema design cannot enforce the constraints, write triggers for that. See an example in the class activity on September 4, 2018.
   - A trigger is worth 5 points

3. (20 points) Provide SQL DDL statements to insert 2 rows of data of your choice into each table that you created as the answer of Q2. Name this script <yournetid>HW2_Q3.sql where <yournetid> is replaced with your actual netid. Make sure that this script executes without any problem on MySQL.
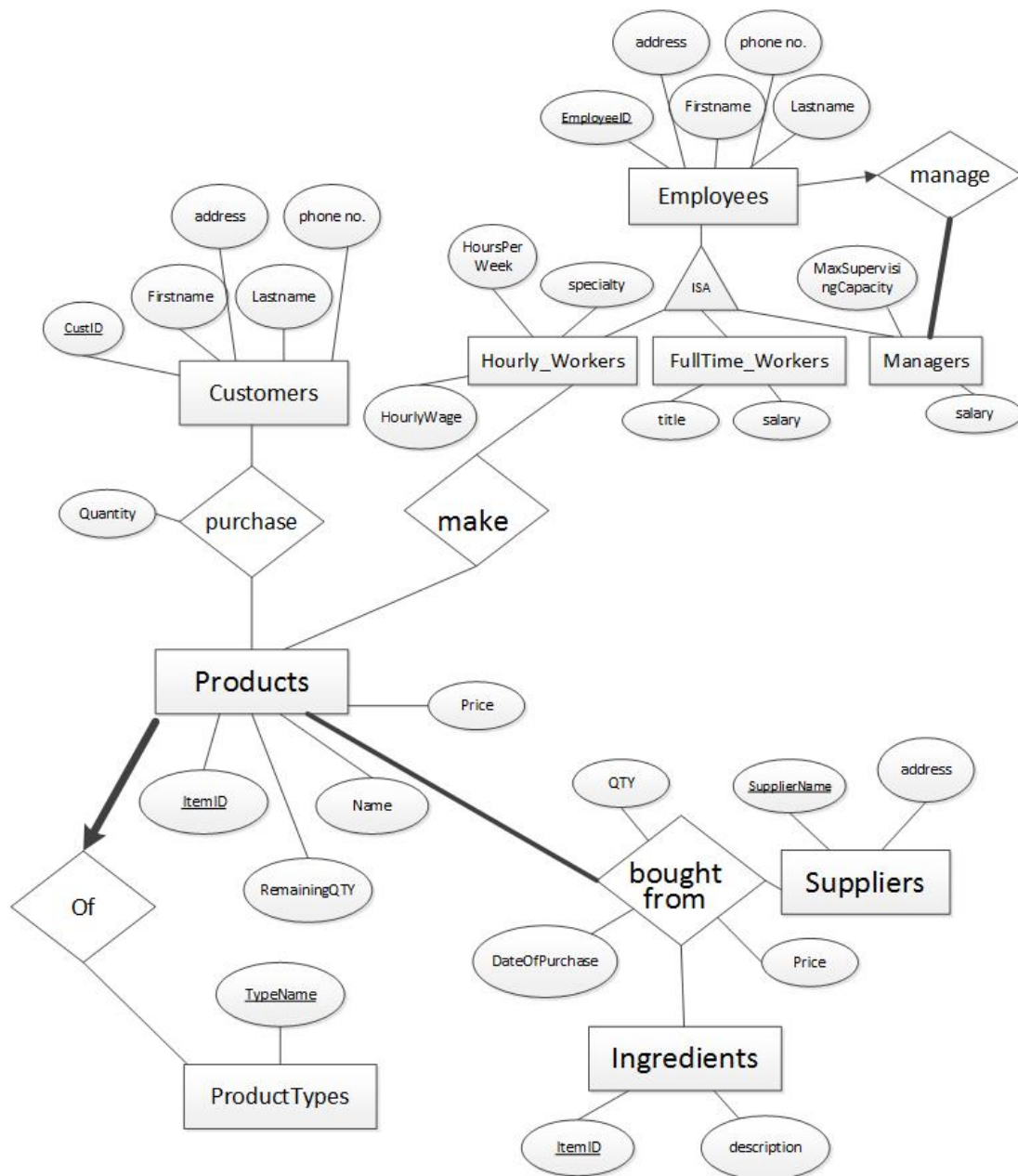
Figure 1: ER diagram of Pak's food factory; note that this diagram is slightly different from the diagram given in Assignment I in that there is no overlapping constraint and no covering constraint for the ISA relationship set. Furthermore, the Get_discount relationship set was removed.

4. (10 points) Consider the following relational schemas. Run HW2_Q3DDL.sql in MySQL workbench to generate the database on your own.

   Suppliers(sid: int, sname: VARCHAR(30), address: VARCHAR(50))
   Parts(pid: int, pname: VARCHAR(30), color: VARCHAR(10))
   Catalog(sid: int, pid: int, cost: double)

The Catalog relation lists the prices charged for parts by Suppliers. Catalog.pid is a foreign key to Parts.pid. Catalog.sid is a foreign key to Suppliers.sid. Write the following queries in SQL. Execute them to make sure you get the correct answers. Copy the SQL queries and include the screenshot of the results as the answers. Do not list other attributes not specified in the return results.

5. (15 points) Write SQL queries to find the information below. Put your queries in <netid>HW2_Q4.sql and capture screenshots after executing the queries. Submit both <netid>HW2_Q4.sql and the screenshots. Replace <netid> with your netid.

   a. (2 points) List sid, sname, and address of all suppliers.
   b. (2 points) List pid, pname for black parts only.
   c. (2 points) Return the number of unique parts.
   d. (2 points) Show all the catalog information indicating which part is supplied by which supplier in ascending order of suppliers.sname. List only the values of these attributes, Parts.pid, Parts.pname, Suppliers.sid, Suppliers.sname, and Catalogs.cost.
   e. (2 points) Find all distinct black parts in the Catalog. Show only the *pid* and *pname* of the black parts.
   f. (2 point) Find distinct *sids* and *snames* of suppliers who supply a red part or a green part.
   g. (1 point) Find distinct sids and snames of suppliers who supply a red part and a green part.
   h. (1 point) Find the snames of suppliers who supply every blue part.
   i. (1 point) Find the pids, pnames, and the maximum cost for that part among all the suppliers.